MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute
Affiliated to RGPV, Bhopal)

Skill Based Mini Project Report

on

"UNIVERSITY MANAGEMENT SYSTEM"

Submitted By:

Shivansh Nigam

0901CS201119

CSE, II Year, 4th Semester

Faculty Mentor:

Ms. Jaimala Jha

Assistant Professor

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Shivansh Nigam** (0901CS201119) has submitted the project report titled **"University Management System"** under the mentorship of **Ms. Jaimala Jha,** in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

Prof. Jaimala Jha
Faculty Mentor
Assistant Professor
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Ms. Jaimala Jha**, **Assistant Professor, Computer Science and Engineering.**

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Shivansh Nigam
(0901CS201119)
II Year, 4th SEM
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering, for allowing** me to explore this project. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of Ms. **Jaimala Jha**, Assistant Professor, Computer Science And Engineering, for her continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Shivansh Nigam
(0901CS201119)
II Year, 4th SEM
Computer Science and Engineering

ABSTRACT

Project "University Management System" , inculcates us with the concept of managing and organising a database.We have used the concept **of GUI to implement** an interface to insert and maintain data in the database, **tkinter** is used for the same. University Management System deals with the maintenance of university, college, faculty, student information within the university. UMS has a relational database, which is used to store the college, faculty, student, courses and information of a college.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

UMS : University Management System

GHz : Giga Hertz

GUI : Graphical User Interface

Tkinter: acronym for “Tk Interface”

INTRODUCTION

UMS makes management to get the most updated information always by avoiding manual accounting process. It operates on a 2 layer architecture :

Administrator

User (students, teachers, courses registered by the students)

University Administrator has the functionality of registering new colleges and courses. He has the rights of creating department, allocating courses to departments, creating faculties, students and allocating subjects to faculties and modifications in the data entered by the user can also be done by the college administrator.

User of this may be faculty or students or department staff. Faculty has the facility of entering the marks and attendance of the students. Students can check their marks and attendance but there is no chance of modifications. Department staff can maintain records respective to their roles.

OBJECTIVE OF THE PROJECT:

The idea behind the whole task was to automatize the procedure of daily affairs of university.

It deals with the maintenance of university, college, faculty, student information within the university. This project of involved the automation of student information that can be implemented in different college managements.

Point of concern:

Some major problems are displayed below:

• Maintaining records in a register is long-winded.

• Reliability of records cannot be guaranteed because of being prone to human errors.

• The recorded data's security cannot be guaranteed as it can be easily altered or theft.

TOOLS USED

HARDWARE ESSENTIALS

- Processor: Minimum 1 GHz; Recommended 2GHz or more.
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
- Hard Drive: Recommended 64 GB or more.
- Memory (RAM): Recommended 4 GB or above

SOFTWARE ESSENTIALS

- Operating system: Windows or MacOs or Linux
- Language: Python(GUI Implementation : tkinter)
- VS code for implementation of code.

IMPLEMENTATION:

- **Some functions/ commands implemented are:**
- Import: to import standard libraries like mysql.connector, tkinter
- cursor() : to adjust cursor in the screen.
- execute() : to execute table creation if not exist
- label() : for naming the grid,
- INSERT : SQL command to insert entries in the respective table
- connect() : implementation of mysql.connector
- CREATE : SQL command to create a schema
- Iconbitmap, etc.

SCHEMA:

- **Table teachers (**

  TID INTEGER,

  teacher_name VARCHAR(30) ,

  COURSE_ID INTEGER **);**

- **Table students(**

  sid INTEGER ,

  STU_name VARCHAR(30),

  COURSE_ID INTEGER **);**

- **Table administrator(**

  aid INTEGER,

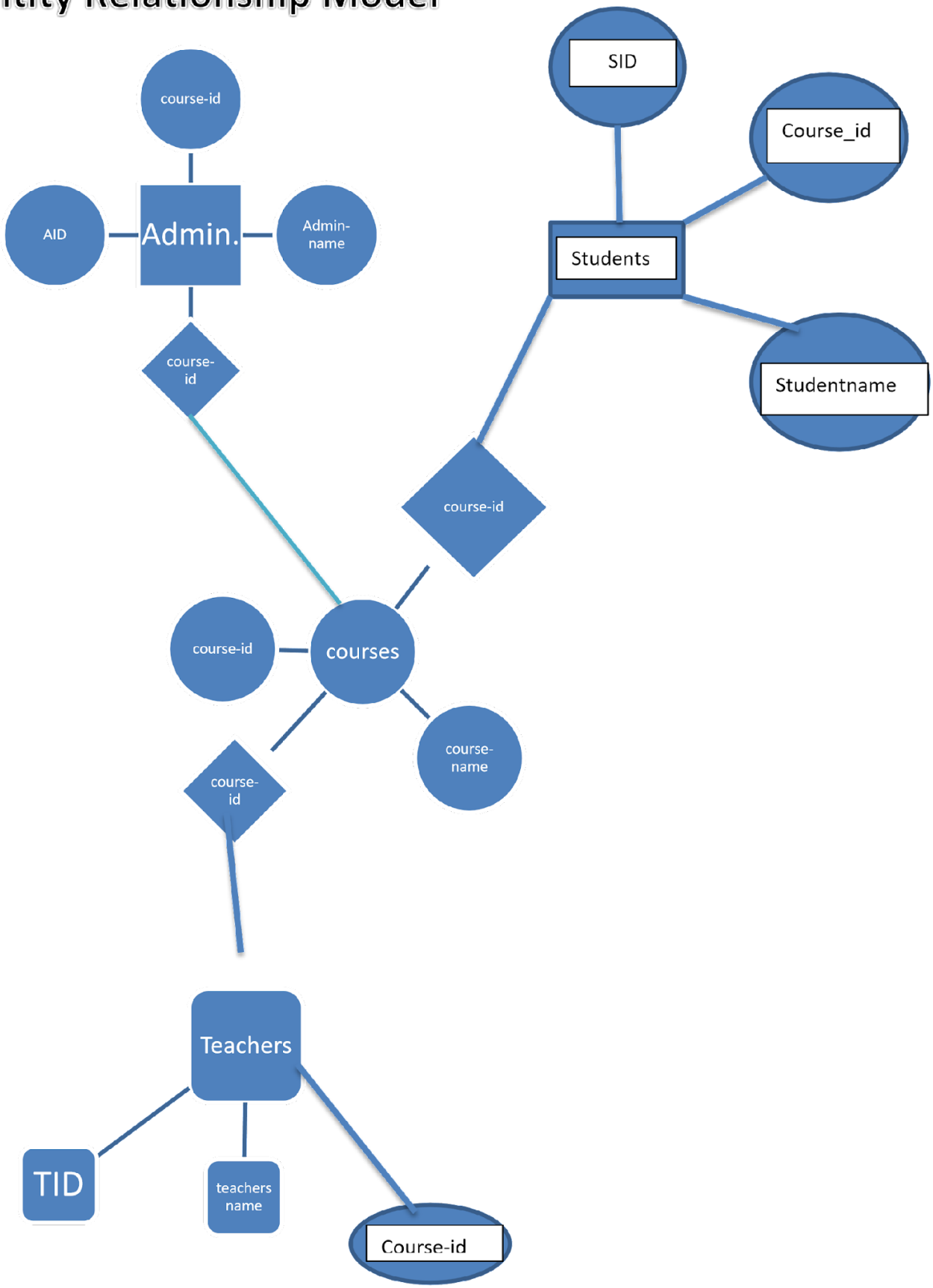
  ADMINISTRATOR_NAME VARCHAR(40),

  COURSE_ID INTEGER**)**;

- **Table courses(**

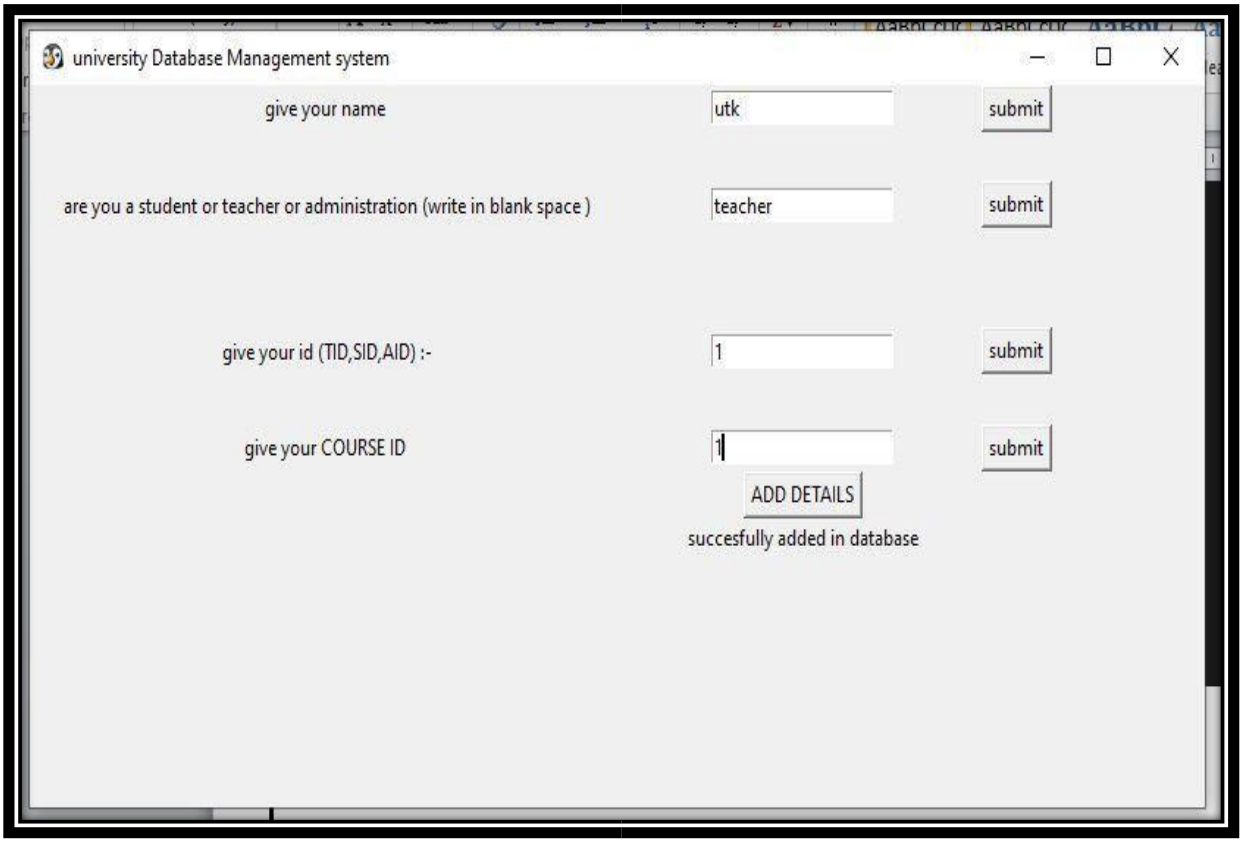
  COURSE_ID INTEGER , course_name VARCHAR(30)**);**

Entity Relationship Model

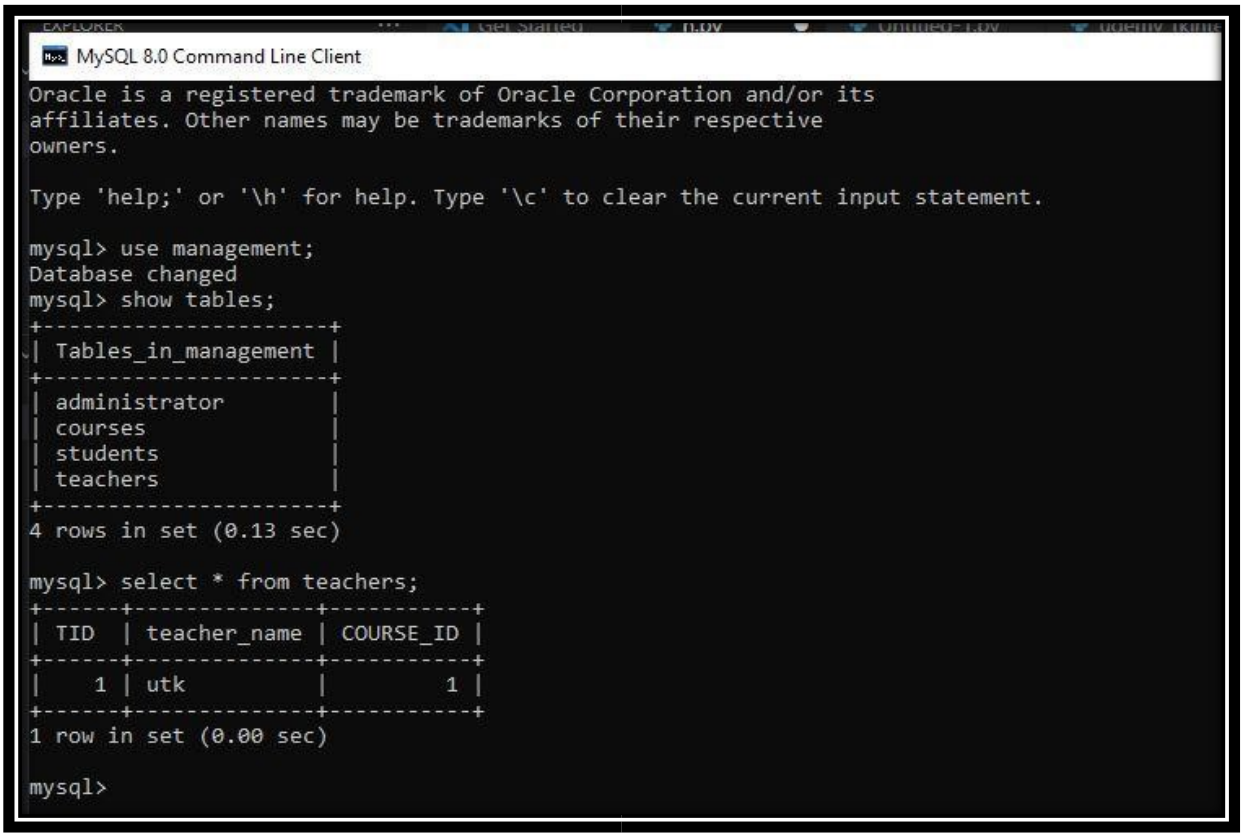


RESULT

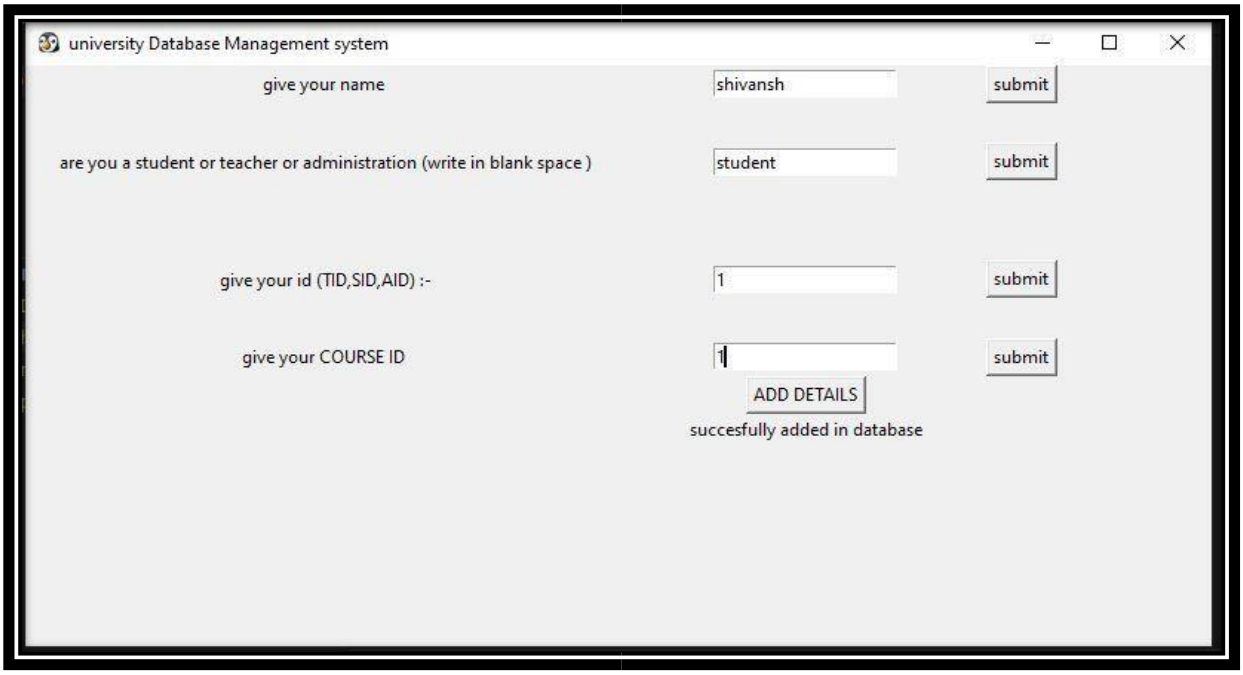
OUTPUT SCREEN :-

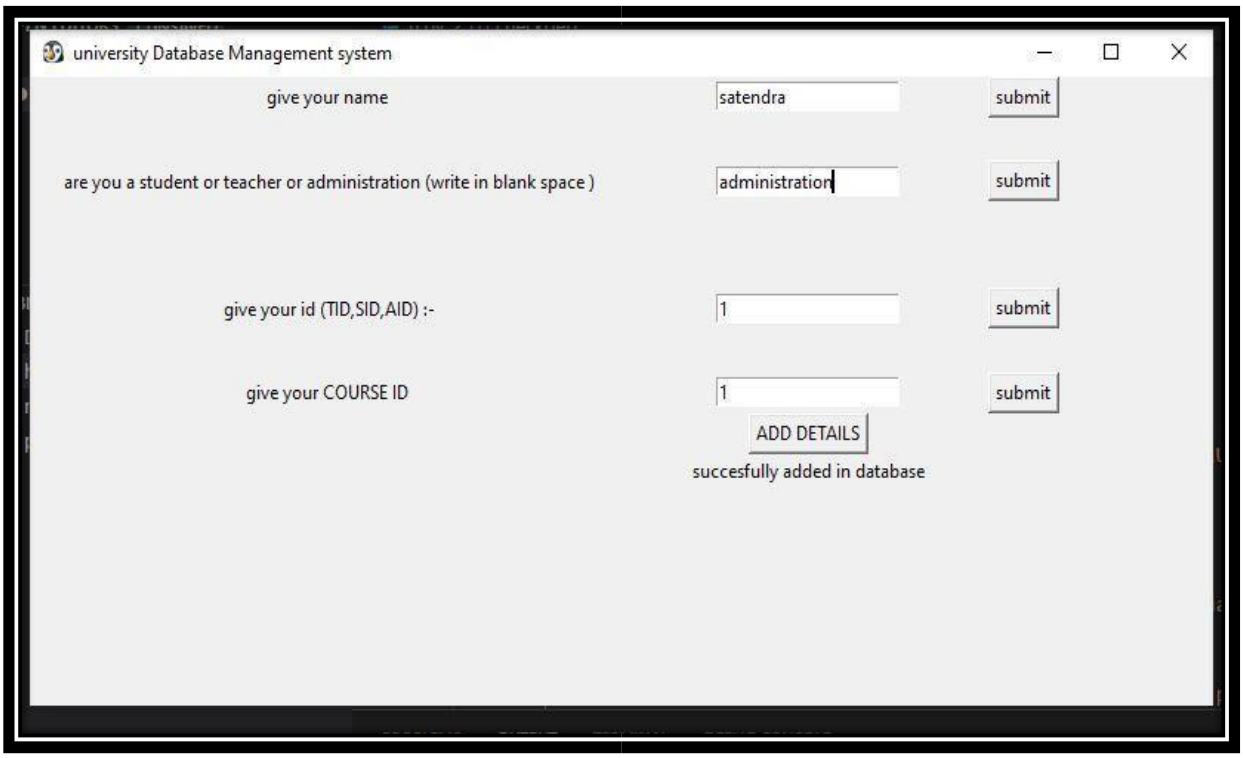


DATABASE:-

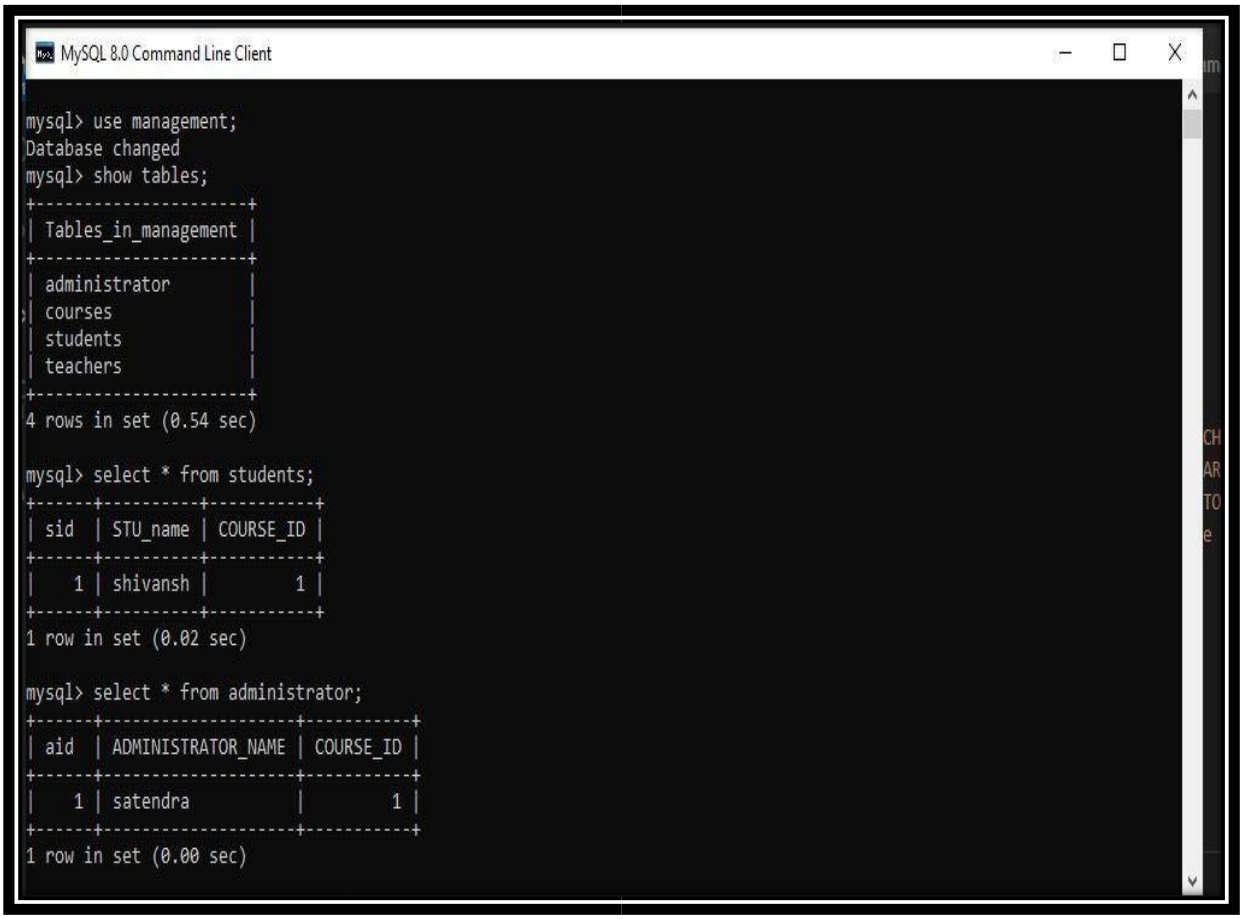

MySQL 8.0 Command Line Client
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> use management;
Database changed
mysql> show tables;
+----------------------+
| Tables_in_management |
+----------------------+
| administrator |
| courses |
| students |
| teachers |
+----------------------+
4 rows in set (0.13 sec)
mysql> select * from teachers;
+------+--------------+-----------+
| TID | teacher_name | COURSE_ID |
+------+--------------+-----------+
| 1 | utk | 1 |
+------+--------------+-----------+
1 row in set (0.00 sec)
mysql>

OUTPUT 2:


university Database Management system
give your name
shivansh
submit
are you a student or teacher or administration (write in blank space)
student
submit
give your id (TID,SID,AID) :-
1
submit
give your COURSE ID
1
submit
ADD DETAILS
succesfully added in database


university Database Management system
give your name
satendra
submit
are you a student or teacher or administration (write in blank space)
administration
submit
give your id (TID,SID,AID) :-
1
submit
give your COURSE ID
1
submit
ADD DETAILS
succesfully added in database

RESULTANT DATABASE AFTER INSERTION :



.

CONCLUSION

After successfully completing the minor project, a simple conclusion can be drawn that an “University Management System” is a well organised management system. The proposed system magnificently maintains the records of this organization. Relational database implementation efficiently inculcates all the essential features whether it is maintenance of records or updation of records. These details are to be entered and retrieved manually, due to which of there are certain disadvantages such as time consumption, inaccuracy of data, updating process, and much more.

SCOPE OF THE PROJECT

This project can be improved in various aspects such as record updation and record entry can be made much less tiresome. Also work should be done on its disadvantages like inaccuracy, time consumption, manual entry, etc. With more advancement in our learning , we can develop this project .
Table management and much more interactive GUI can be fields of improvement. Also concern of data security and redundancy must be rectified.

.

REFERENCES:

- Github for viewing other project as reference
- W3schools for efficiently learning tkinter implementation.
- Mysql connector implementation tutorials
- Studocu

.

APPENDICES

The following is the code :

```
import mysql.connector as c

from tkinter import *

mydb =c.connect(

  host = "localhost" ,

  user = "root",

  password ="Shivansh@1",database= "management"
  )
mycursor = mydb.cursor()

mycursor.execute("create table if not exists teachers(TID INTEGER,
teacher_name VARCHAR(30)  , COURSE_ID integer)")
 mycursor.execute("create table if not exists students(sid integer  , STU_name
VARCHAR(30),COURSE_ID integer)")
 mycursor.execute("create table if not exists administrator(aid integer ,
ADMINISTRATOR_NAME VARCHAR(40) , COURSE_ID integer)")

mycursor.execute("create table if not exists courses(COURSE_ID integer ,
course_name VARCHAR(30))")

root = Tk()

#inp = Entry(root,width= 50)

root.geometry("800x400")
 root.minsize(400,400) root.maxsize(2000,2000)

root.title("university Database Management system")

root.iconbitmap("penguin.ico")

root.resizable(1,1)

root.resizable(1,1)

E1 = Entry(root,width =20 )
E1.grid(row =0,column=1)
```

```
E2 = Entry(root,width =20 )
E2.grid(row =4,column=1)

E3 = Entry(root,width =20 )
E3.grid(row =8,column=1)

E4 = Entry(root,width =20 )
E4.grid(row =12,column=1)
name=
E1.get()
dep =
E2.get()
pid =
E3.get()
cid =
E4.get()
 def
ADD_DET
AILS():

      l5 = Label(root,text ="succesfully added in database")

      l5.grid(row =  15 , column = 1,padx = 40)
 def
checkd
ep():
      val = (pid,name,c)

      sp = "INSERT INTO teachers (TID,teacher_name,COURSE_ID)
      VALUES (%s,%s,%s)"
               if
dep=="teacher" :

            mycursor.execute(sp,val)
              elif
dep=="student":

            mycursor.execute("INSERT INTO students

            VALUES(pid,name,cid)")
                  elif dep==
"administration":

            mycursor.execute("INSERT INTO administrator
VALUES(pid,name,cid)")
```

```
l1 = Label(root, text = 'give your name ' )

l1.grid(row=0,column=0, padx = 20, pady=0  )

l2 = Label(root,text = "are you a student or teacher or administration (write in
blank space )")

l2.grid(row = 4,column =0,padx= 20 ,pady =30)

b1 = Button(root,text = "submit") b1.grid(row =
0,column =2)

b2 = Button(root,text = "submit",command= checkdep) b2.grid(row =4,column
=2 )

l3 = Label(root, text = 'give your id (TID,SID,AID) :-' )
l3.grid(row=8,column=0, padx = 20, pady = 30)

b3 = Button(root,text = "submit") b3.grid(row
=8,column =2 )

l4 = Label(root, text = 'give your COURSE ID ' ) l4.grid(row=12,column=0,
padx = 20 )
 b4 = Button(root,text = "submit")
b4.grid(row = 12,column = 2)

b5 = Button(root,text="ADD DETAILS",command = ADD_DETAILS)
b5.grid(row = 14,column = 1)

root.mainloop()
```